UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France

Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1276

Programme 3
Réseaux et Systèmes Répartis

MODELS FOR COMMUNICATIONS IN THE HYPERCUBE

Philippe JACQUET
Paul MUHLETHALER

Août 1990

RR-1276

Models for Communications in the Hypercube

Philippe Jacquet and Paul Mühlethaler

INRIA Rocquencourt. Domaine de voluceau, BP 105
78153 Le Chesnay Cedex

Abstract : In this paper we study a very simple model of communication in the hypercube. With the assumptions of a symmetrical Poisson traffic, we develop two analytical models to compute the maximum achievable load and mean transmission delay of a packet. The model is also simulated and the matching between analytical models and results of simulation is good.

Modèles pour les communications dans l'hypercube

Résumé : Dans ce papier nous étudions un modèle très simple de communication dans l'hypercube. Avec des hypothèses d'un trafic symétrique et de Poisson, nous developpons deux modèles analytiques pour calculer le débit maximum atteignable et le temps moyen de transmission d'un paquet. Le modèle est également simulé et la correspondance entre les modèles analytiques et les résultats de simulation est bonne

I. INTRODUCTION

Interconnexion networks are extensivety used in supercomputers to ensure efficient communication between processors which are working to the achievement of a same program.Such networks like bayan,omega have been extensivly studied. One can find interesting results in [1] [2]. The hypercube is also a solution to create network of processors working together. Our aim here, is to give simple models to evaluate the performance of such a system.

Starting from very simple and natural hypothesis, we adress the problem of available bandwidth and average transmission delay of packets.

Results of mathematical models are compared with those obtained out of simulations. The following is divided into three parts. In the first one, we describe the structure of the hypercube and the model for the trafic between processors. In the second one, we develop mathematical models. In the third one we compare results obtained out of simulations with results of mathematical models. In the appendix, one can find intermediate computations not immediatly necessary to understand the paper.

II. HYPERCUBE

Most of the interconnection networks in supercomputers are existing independently from processors they are connecting. Such networks have therefore their own hardware, processors connected by such networks are not involved in the communication process [3],[4].

In the hypercube, every processor is a node of the network and must participate to the communication process even for packets it has not issued. Packets are forwarded on each node until they reach their own destination. The hypercube consist of $N=2^n$ processors, each of them has n neighbours. We can give each processor a binary address composed of n bits 0 or 1. Each processor is physically connected with processors which have the same adress except for one bit.

A packet is sent by a station with a destination address. The packet is forwarded from neighbouring station to neighbouring station until it has reached its destination. A given packet has in general more than one possible path to reach its destination. A processor will therefore choose the next processor at random among all the processors on a possible path.

It is the same if this processor had already received a packet. We can easily notice that more than one packet can be waiting to be forwarded in a processor. The first packet arrived will be the first forwarded.

III. ASSUMPTIONS AND MATHEMATICAL MODELS

First of all, we will use a slotted model. At each slot, each processor can send or (exclusive) receive a packet.If, for example, a given processor has already sent a packet, other processsors won't be allowed to send it a packet. Symmetrically a processor which has already received a packet can not send a packet.

We easily notice that more than one packet can be waiting to be forwarded in a processor. Packets in a processor will be forwarded FIFO, the first arrived will be the first forwarded ; new packets generated in a processor follow this same rule than forwarded packets. We suppose that the storage capacity of a processor is infinite .

The traffic generated by processors is Poisson of rate λ packets per slot and per node, this traffic is symmetrical. For a packet generated at a given node its destination is fairly distributed among one of the other processors.

The model described here is based on the derivation of the probability that a node is busy *ie* sends or receives a packet.

III.1. Basic equations

Since the destination of the generated packets in a station is chosen at random among the other processors, the probability that a packet has to do k hops exactly before being delivered is :

$$\frac{2^{-n}}{1-2^{-n}}\binom{n}{k}$$

The probability p that a node is blocked is given by the following formula :

$$p = \lambda\left(mean\ number\ of\ emissions\ per\ packet + (1 - \frac{1}{n})\ mean\ number\ of\ receptions\ per\ packet \right)$$

The correcting term (1-1/n) comes from the fact that a node which receives a packet is blocked for every neighbour node except for the node which has sent the packet. The mean number of emissions is equal to the mean number of receptions and is also equal to the mean distance between two different nodes.

$$d = \frac{2^{-n}}{1-2^{-n}} \sum_{k}^{n} k \binom{n}{k} = \frac{n}{2} \quad \frac{1}{1-2^{-n}} \cdot$$

Therefore we have :

$$p = \frac{\lambda}{1-2^{-n}} (n - \frac{1}{2}).$$

Let us introduce X_k :

$$X_k = \frac{1}{1-p^k} + \frac{1}{1-p^{k-1}} + ... + \frac{1}{1-p}$$

X_k is the mean service time used by a packet generated at distance k from its destination (with k hops to perform). Therefore

$$\sum_{k=1}^{n} \frac{2^{-n}}{1-2^{-n}} \binom{n}{k} X_k = X$$

is the mean service time used by a random packet. The stability of the system requires :

$$\lambda X \leq 1.$$

The maximum available bandwidth is therefore λ_{max}:

$$\lambda_{max} X = 1 \quad with \quad p = \frac{\lambda}{1-2^{-n}} \left(n - \frac{1}{2}\right).$$

Our aim now is to derive the mean time delay between the arrival of a packet in a processor and its arrival in its destination processor. We have two models the first and the more simple is based on a Poisson approximation ; the second one considers a Bernoulli arrival at each slot.

III.2 Average delay with Poisson approximation

We will assume that at every given node the mixture of the traffic locally generated and the traffic of packets transiting through the node is a Poisson traffic.
The input rate of packets passing through a random node is.

$$\frac{\lambda n}{2(1-2^{-n})}$$

This rate includes packets generated at the node and packets passing through the node.
Among these packets, there are :

$$\lambda \sum_{k'=k}^{n} \frac{2^{-n}}{1-2^{-n}} \binom{n}{k'},$$

packets which are passing by the distance k. Therefore the probability v_k to find a packet passing by the distance k is :

$$v_k = \frac{\sum_{k'=k}^{n} 2^{-n} \binom{n}{k'}}{n/2}.$$

The second moment of the service time needed for a packet at distance k to reach distance k-1 is :

$$\frac{1+p^k}{(1-p^k)^2}.$$

Therefore the second moment of the service time is :

$$\frac{\sum_{k=1}^{n} \sum_{k'=k}^{n} 2^{-n} \binom{n}{k'} \frac{1+p^k}{(1-p^k)^2}}{\frac{n}{2}}.$$

Let us define :

$$Q_k = \frac{1+p}{(1-p)^2} + \ldots\ldots + \frac{1+p^k}{(1-p^k)^2} \quad Q = \sum_{k=1}^{n} 2^{-n} \binom{n}{k} Q_k \ .$$

Each node is supposed to be a M/G/1 queue. The input on each node is $n\lambda/2(1\text{-}2^{-n})$, this includes packets hopping from node to node. The mean service time for a packet to reach the next node is:

$$\frac{2X(1-2^{-n})}{n}.$$

The second moment of this service time is according to previous computations

$$\frac{2Q(1-2^{-n})}{n}.$$

We can now apply the well known formulas concerning the M/GI/1 queue.The mean delay in queue is:

$$W = \frac{\dfrac{n\lambda}{4(1-2^{-n})}\ \dfrac{2Q(1-2^{-n})}{n}}{1-\dfrac{\lambda n}{2(1-2^{-n})}\ \dfrac{2X(1-2^{-n})}{n}} = \frac{\dfrac{\lambda}{2}Q}{1-\lambda X}.$$

The mean delay between the generation of a random packet and its arrival at the destination processor is :

$$T = \frac{n}{2(1-2^{-n})}\left(\frac{2X(1-2^{-n})}{n} + \frac{\dfrac{\lambda}{2}Q}{1-\lambda X}\right).$$

It comes :

$$T = X + \frac{n\lambda}{4(1-2^{-n})}\ \frac{Q}{1-\lambda X}.$$

III.2 Average delay with Bernoulli model

At each node we decompose the traffic in packets in transit (this traffic can be modeled as a Bernoulli traffic) and the packets generated locally, this traffic is Poisson. Let us introduce the following notations :

v_k the probability for a random packet to hop from distance k to distance k-1.

μ the rate of the Bernoulli traffic.

It is easy to see that the generating function of the time spent by a random packet before hopping to the next node is :

$$\sum_{k=1}^{k=n} \nu_k \frac{(1-p^k)z}{1-p^k z}.$$

The generating function of arrivals during two hops of a random packet (outside arrival plus packets hopping from node to node) is :

$$A(z) = \sum_{k=1}^{\infty} \nu_k \frac{(1-p^k)e^{\lambda(z-1)}}{1-p^k(1-\mu(1-z))e^{\lambda(z-1)}}.$$

Let $q(z)$ be the generating function of the length of the queue at each node. $q(z)$ satisfies the following equation :

$$q(z) = A(z)\frac{q(z)-q_0}{z} + q_0(1-\mu(1-z))e^{\lambda(z-1)}.$$

Renewal points are the times when the queue is empty or after a service.

It comes :

$$q(z) = q_0 \frac{A(z) - z(1-\mu(1-z))e^{\lambda(z-1)}}{A(z)-z}.$$

In the appendix, we give all the needed computations to get q_0 and $q'(1)$. In particular, it will be necessary to compute the expansion at the second order of $A(z)$. The results of these computations give :

$$q_0 = \frac{1+\mu-(\lambda+\mu)\dfrac{2X(1-2^{-n})}{n}}{1-(\lambda+\mu)\dfrac{2X(1-2^{-n})}{n}+2\mu+\lambda}.$$

The condition $q_0 = 0$ is equivalent to :

$$X = \frac{1+\mu}{\lambda+\mu}\,\frac{n}{2(1-2^{-n})}.$$

The Bernoulli rate μ is equal to :

$$\mu = \frac{\lambda\dfrac{n}{2}(1-2^{-n})-\lambda}{1-\lambda\dfrac{n}{2}(1-2^{-n})}.$$

Transiting packets arrive at slots where no packets are allowed to exit (every $(1-\lambda n(1-2^{-n})/2)$ slots)
The condition becomes :

$$\lambda X = 1,$$

which is the same condition of saturation as in the Poisson model.

It is then possible to compute $q'(1)$; we have :

$$q'(1) = \frac{(1-q_0)\left(\frac{\lambda^2}{2} + \lambda\mu)(X-1) + (\lambda+\mu)^2 \sum_{k=1}^{\infty} \frac{v_k p^{2k}}{(1-p^k)^2} + \frac{\lambda^2}{2}\right) + q_0(\lambda + \mu + \lambda\mu)}{1 - \lambda - (\lambda+\mu)(X-1)}.$$

According to the theory of renewal processes, the mean number of packets in a queue is :

$$M = \frac{q'(1)X + \frac{\lambda}{2}\left((1-q_0)Q + q_0\right) + \frac{\mu}{2}(1-q_0)\left(Q - X\right)}{X(1-q_0) + q_0}.$$

According to the Little formula,we have the average delay for a random packet:

$$D = \frac{2M}{\lambda n(1-2^{-n})}.$$

III MATCHING BETWEEN RESULTS OF SIMULATION AND ANALYTICAL MODELS.

The simulation software is written in C^{++} and uses Sphinx an event driven simulator developped at INRIA. To prevent fairness in bandwidth allocation the software is in charge of doing a random permutation which selects at each slot the processor which first tries to forward its packets.

We have investigated three hypercubes corresponding to $N=16$, $N=32$ and $N=64$. The matching between simulation results and results of the analytical model is very good.

IV CONCLUSION

We have presented a very simple model for communication in the hypercube based on reasonable assumptions. Two analytical models are developped to compute the maximum achievable load with the hypothesis of a symmetric Poisson traffic. Our model is simulated with Sphinx an event driven simulator and the matching between simulation results and results obtained out of the analytical models is good. Closed formulas given by the analytical models could be good tools for people designing architectures based on the hypercube.

RÉFÉRENCES

[1] Analysis and Simulation of Buffered Delta Networks. Daniel M. Dias, J. Robert Jump. IEEE transactions on Computers. Vol C 30 N° 4. April 81

[2] Design and Performance of Generalized Interconnection Networks. Laximi N. Bhujon Dharma P. Agrawal. IEEE Transactions on Computers Vol C 32. N°12. December 83.

[3] T. Y. Feng "A survey of Interconnection Networks" Computer; Dec 81 pp 12-27

[4]H. J. Siegel "Interconnection Network for Large Scale Parallel Processing : Theory and Case studies, Lexington Books, Lexington, Mass. ,1985

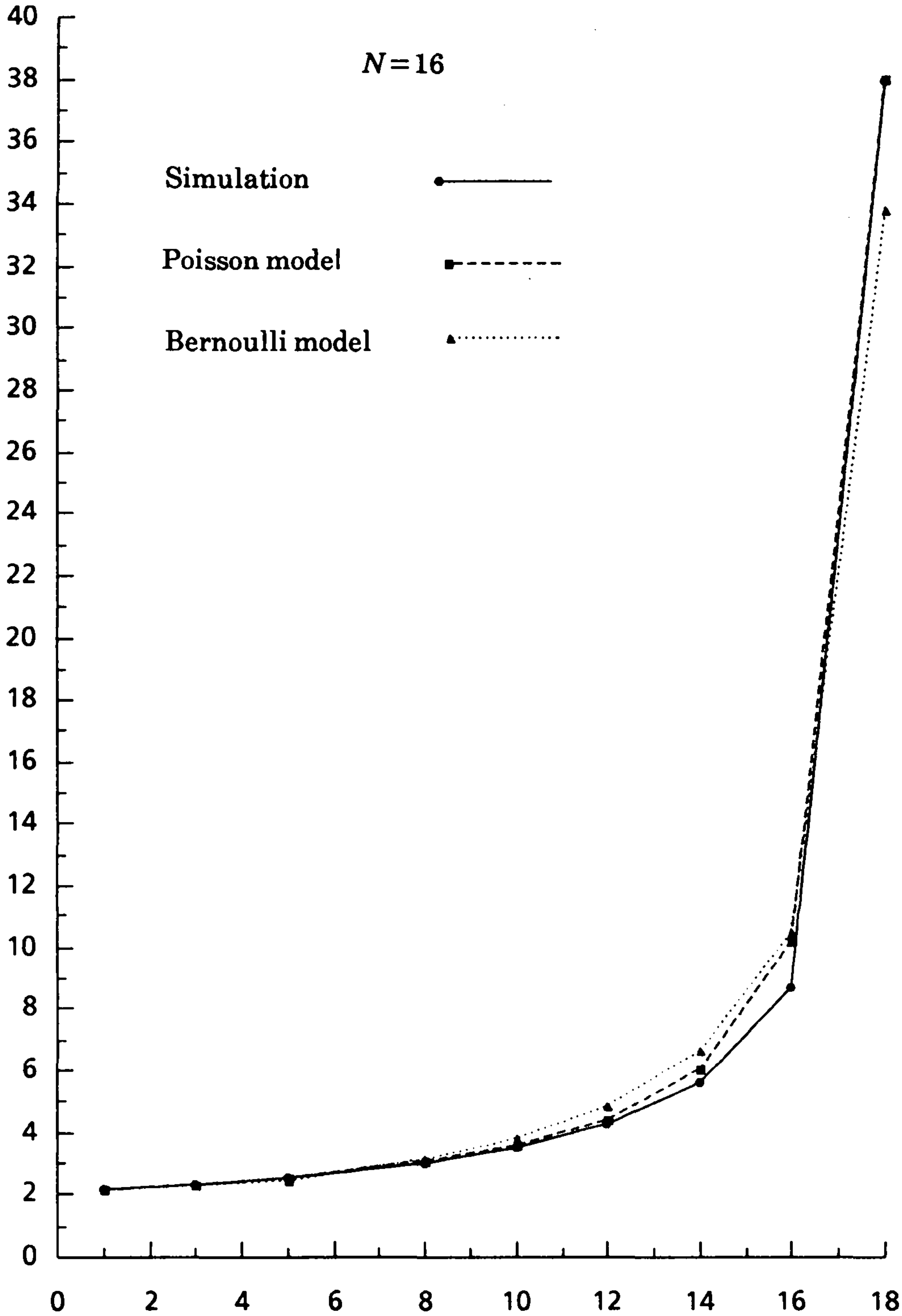

Fig 1.Mean delay versus input load in percent of packets per node and per slot for $N=16$.

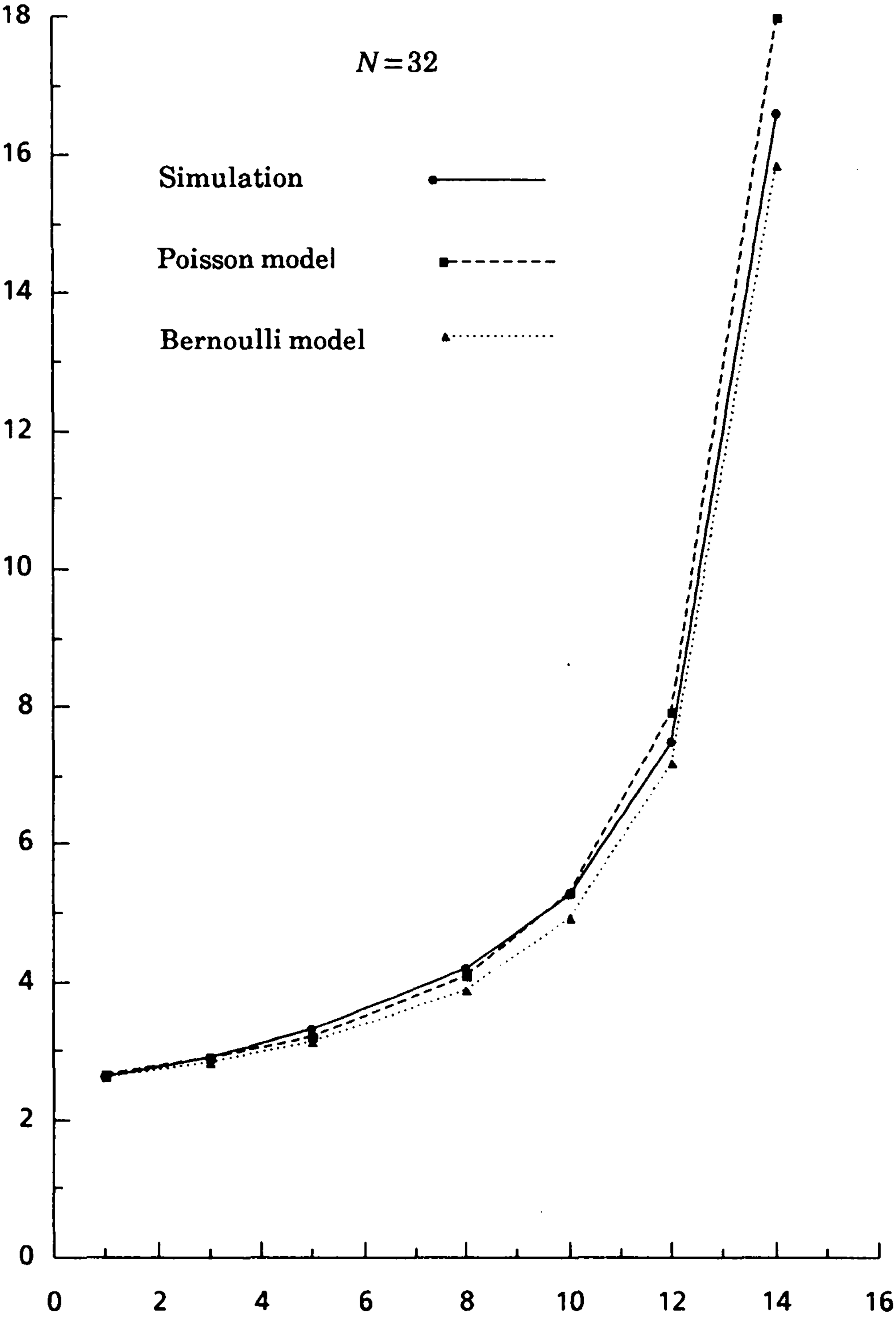

Fig 2.Mean delay versus input load in percent of packets per node and per slot for $N=32$.

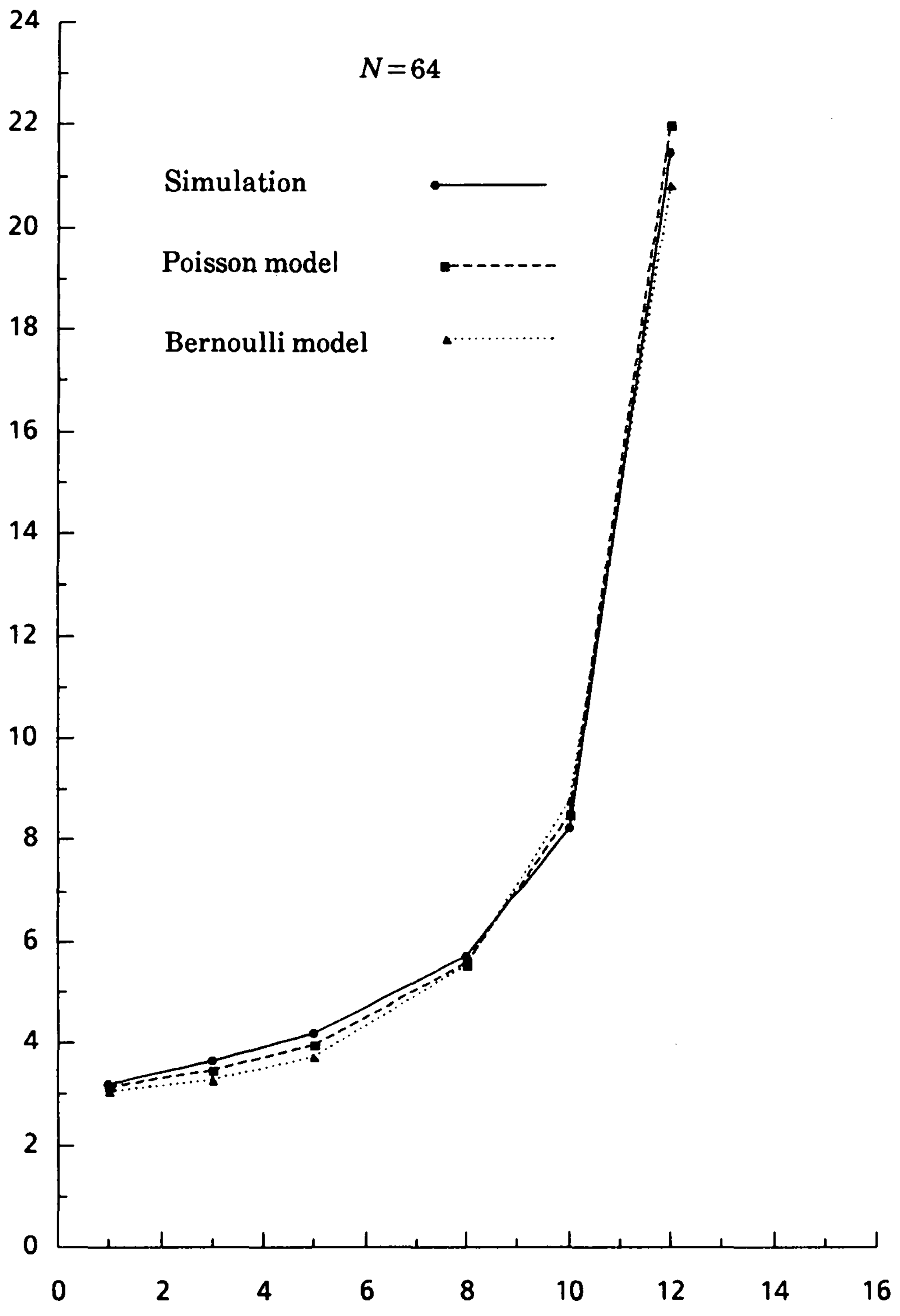

Fig 3.Mean delay versus input load in percent of packets per node and per slot for $N=64$.

APPENDIX

Derivation of q_0 and $q'(1)$.

First, we give the expansion at the second order of $A(z)$ in the neighbourhood of $z=1$:

$$A(z)=1+\left(\lambda+(\lambda+\mu)(\frac{2X(1-2^{-n})}{n}-1)\right)(z-1)+\left(\frac{\lambda^2}{2}+\left(\frac{3\lambda^2}{2}+2\lambda\mu\right)(\frac{2X(1-2^{-n})}{n}-1)\right.$$

$$\left.+(\lambda+\mu)^2\sum_{k=1}^{\infty}\frac{v_k p^{2k}}{(1-p^k)^2}\right)(z-1)^2+o\left((z-1)^2\right)$$

$$q(z)=q_0\frac{1-z+\left(\lambda+(\lambda+\mu)(\frac{2X(1-2^{-n})}{n}-1)\right)(z-1)+\mu z(1-z)\left(1+\lambda(z-1)\right)+o\left((z-1)\right)}{1-z+\left(\lambda+(\lambda+\mu)(\frac{2X(1-2^{-n})}{n}-1)\right)(z-1)+o\left((z-1)\right)}$$

$$1=q_0\frac{\lambda+(\lambda+\mu)(\frac{2X(1-2^{-n})}{n}-1)-\mu-1-\lambda}{\lambda+(\lambda+\mu)(\frac{2X(1-2^{-n})}{n}-1)-1}$$

$$q_0=\frac{1+\mu-(\lambda+\mu)\frac{2X(1-2^{-n})}{n}}{1+2\mu+\lambda-(\lambda+\mu)\frac{2X(1-2^{-n})}{n}}$$

$$q(z)-1=\frac{(q_0-1)A(z)-z\left(q_0-1+\mu z(z-1)e^{\lambda(z-1)}\right)}{A(z)-z}$$

$$\frac{q(z)-1}{z-1}=\frac{(q_0-1)A(z)-z(q_0-1)+\mu z(z-1)q_0e^{\lambda(z-1)}}{(A(z)-z)(z-1)}$$

$$\frac{q(z)-1}{z-1}=\frac{(q_0-1)(A(z)-z)+\mu z(z-1)q_0(1+\lambda(z-1)+\frac{\lambda^2}{2}(z-1)^2+o((z-1)^2))}{(A(z)-z)(z-1)}$$

$$\frac{q(z)-1}{z-1}=\frac{(q_0-1)\left(\frac{\lambda^2}{2}+\left(\frac{3\lambda^2}{2}+2\lambda\mu\right)\left(\frac{2X(1-2^{-n})}{n}-1\right)+(\lambda+\mu)\sum_{k=1}^{\infty}\frac{\nu_k p^{2k}}{(1-p^k)^2}\right)}{\left(1-\lambda-(\lambda+\mu)\left(\frac{2X(1-2^{-n})}{n}-1\right)\right)}$$

$$\frac{+q_0(\lambda+\mu+\lambda\mu+o((z-1)^2))}{\left(1-\lambda-(\lambda+\mu)\left(\frac{2X(1-2^{-n})}{n}-1\right)\right)}$$

$$q'(1)=\frac{(1-q_0)\left(\left(\frac{3\lambda^2}{2}+2\lambda\mu\right)\left(\frac{2X(1-2^{-n})}{n}-1\right)+(\lambda+\mu)^2\sum_{k=1}^{\infty}\frac{\nu_k p^{2k}}{(1-p^k)^2}+\frac{\lambda^2}{2}\right)+(\lambda+\mu+\lambda\mu)q_0}{\left(1-\lambda-(\lambda+\mu)\left(\frac{2X(1-2^{-n})}{n}-1\right)\right)}$$

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399